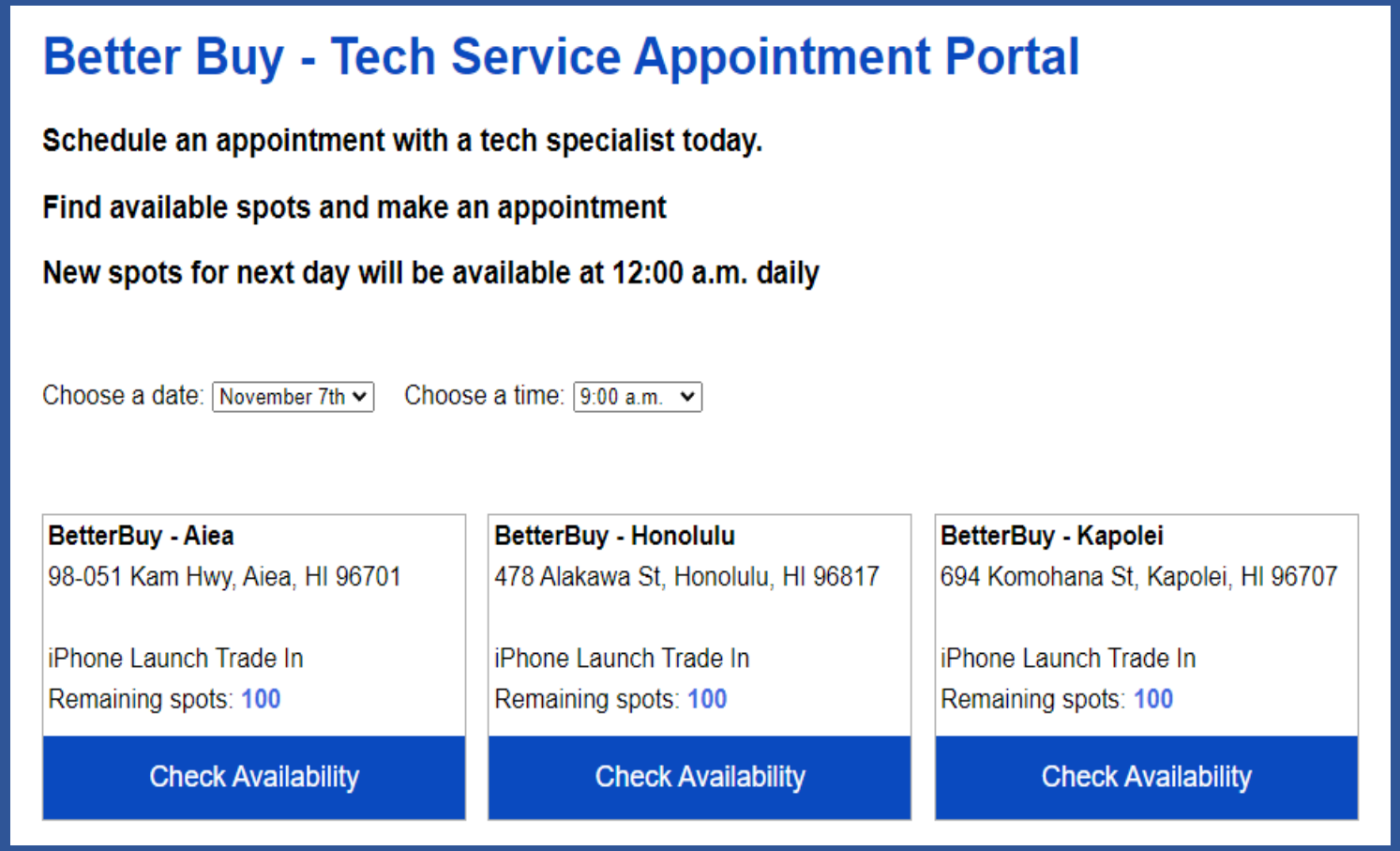
High Volume Retail Appointment Scheduling System



SHOPPING BY APPOINTMENT HELPS RETAILERS REOPEN SAFELY

BY [MARK BURSTEIN] | 06/01/2020

This is a guest blog post from NGC Software's Mark Burstein as part of a series on innovation concepts relevant to retail.

Retailers that were required to close are now beginning to reopen across the U.S., and they are taking great care to ensure shopper safety and put customers at ease as they start shopping again. One way that is getting increased attention from retailers is "shopping by appointment." It's a familiar concept, and it can be the key to helping retailers manage customer traffic.

SCHEDULING AN APPOINTMENT: A FAMILIAR CONCEPT TO CONSUMERS

Consumers make appointments all the time: restaurant reservations, doctor visits and hair appointments are just a few examples. It's a universally accepted concept. In the same way, why can't retailers start scheduling shopping appointments via their apps and websites? This permits a limited number of customers to enter the store on a set schedule while preventing long lines and overcrowding at the store entrance.

Scheduling is much safer and far more efficient than waiting outside a store in long lines. The principles of social distancing have been introduced at many locations, but it's burdensome for shoppers.

Prescheduled shopping appointments allows retailers to apply the same principles, while offering a better customer experience that prioritizes customer safety and ensures a secure, uncrowded shopping environment. Retailers can limit each shopper's visit to a pre-set amount of time, in order to make sure the appointment schedule remains on track.

Select store and time

Drop-off Service for iPhone

Stores near Aiea, HI (96701) Update Location **Sort by** ○ Closest Location ◉ Next Availability

1 Authorized Service Provider

Honolulu (Store 763) 6.0 mi

478 Alakawa St
Honolulu, HI 96817

Friday, November 12

Change Date | 4:40 p.m. – 5:00 p.m. (HST) | Schedule

Follow sign-in prompt on next screen.

Schedule Retail Appointment

Schedule an appointment at an AT&T store and speak to the first available sales representative!

Make Appointments

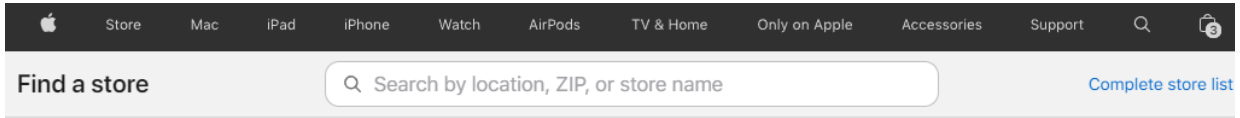
Reserve time for hands-on help.

Make an Appointment

Change Appointments

If your plans change, just reschedule or cancel your appointment.

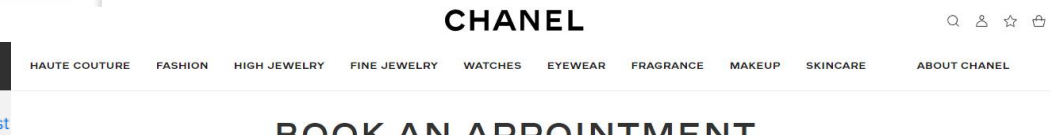
Change an Appointment



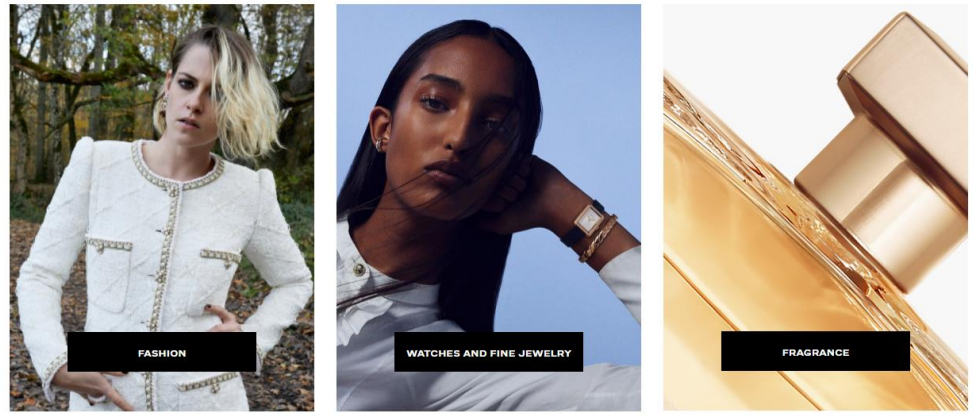
Apple Ala Moana

Opens at 9:30 a.m.

We're open and look forward to welcoming you. Shop by walking in, reserving a one-on-one session with a Specialist or buy online and pickup in store. Get support at the Genius Bar in store or by making a reservation. Face masks required. If you need one, just ask.



BOOK AN APPOINTMENT



1/2

High Volume Retail Appointment Scheduling System

Better Buy - Tech Service Appointment Portal

Schedule an appointment with a tech specialist today.

Find available spots and make an appointment

New spots for next day will be available at 12:00 a.m. daily

Choose a date: November 7th Choose a time: 9:00 a.m.

BetterBuy - Aiea	BetterBuy - Honolulu	BetterBuy - Kapolei
98-051 Kam Hwy, Aiea, HI 96701	478 Alakawa St, Honolulu, HI 96817	694 Komohana St, Kapolei, HI 96707
iPhone Launch Trade In	iPhone Launch Trade In	iPhone Launch Trade In
Remaining spots: 100	Remaining spots: 100	Remaining spots: 100
Check Availability	Check Availability	Check Availability

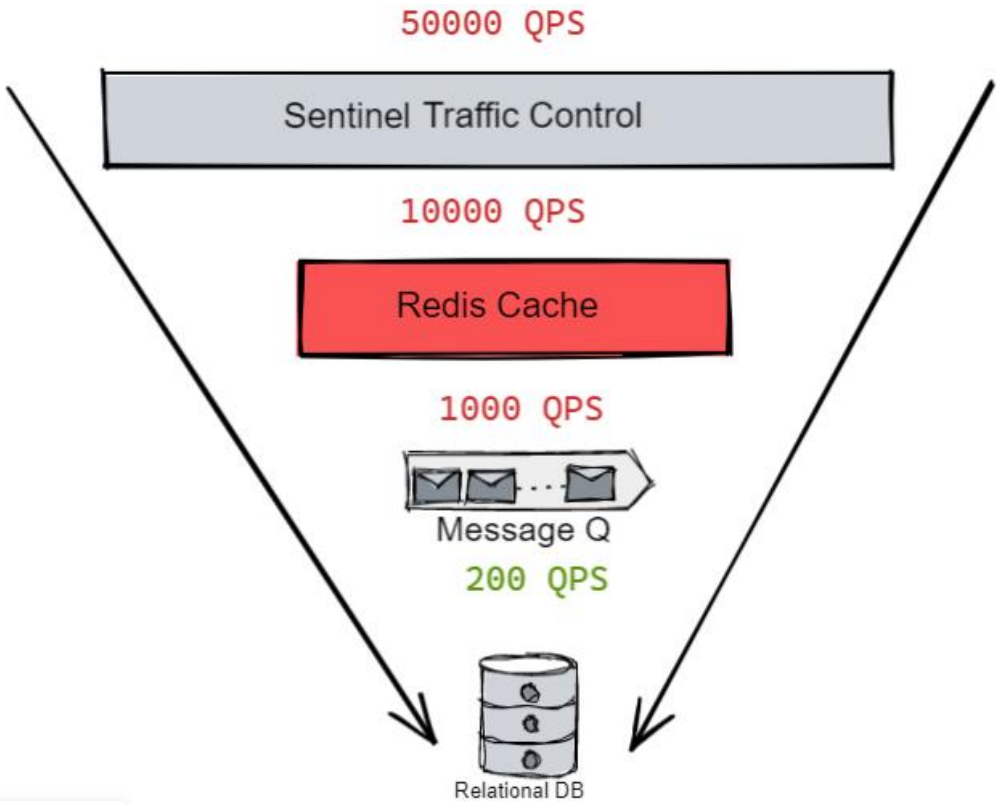
System Requirements:

1. High concurrency and availability
2. Avoid over scheduling

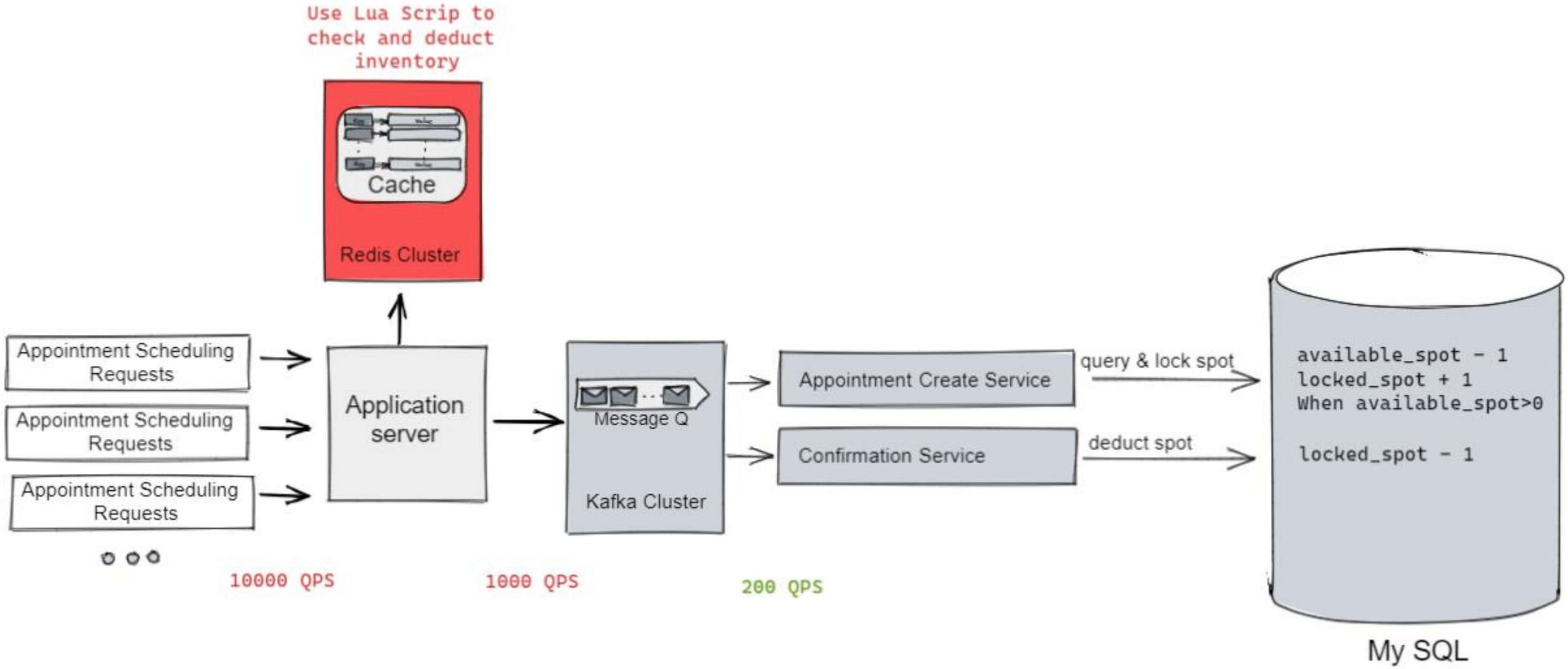
Project Demo

Techniques to Handle High Volume

- Use throttling framework **Sentinel** to perform real time traffic shaping
- **Cache(Redis) warm-up** with Appointment Inventory Info
- Use message queue(**Kafka**) to make process **asynchronous** and achieve **peak load shifting**



Avoid Appointment Over Scheduling



Jmeter Stress Testing

Key Technical Components

- **DB Optimistic Locking** -> prevent over scheduling in Database
- **Redis with Lua Scripting** -> avoid expensive DB queries, atomic appmt locking
- **Message queue**(**Kafka**) -> asynchronize and handle traffic spikes
- **Snowflake Algorithm** -> unique, sorted by time in distributed system
- **Sentinel** -> traffic shaping and system load protection

Database Optimistic Locking

```
<update id="lockSpot" parameterType="java.lang.Integer">
  update appointment_inventory
  set available_spot = available_spot - 1,
      locked_spot = locked_spot + 1
  where id = #{id,jdbcType=INTEGER}
    and  available_spot > 0
</update>
```

Prevent over scheduling.

Database can handle 1000 QPS among all the application it supports. Not enough.
While Redis can handle 100000s request per second.

Redis with Lua Script

```
public boolean spotDeductValidator(String key) {
    try(Jedis client = jedisPool.getResource()) {
        String script = "if redis.call('exists',KEYS[1]) == 1 then\n" +
                "                local spot = tonumber(redis.call('get', KEYS[1]))\n" +
                "                if( spot <=0 ) then\n" +
                "                   return -1\n" +
                "                end;\n" +
                "                redis.call('decr',KEYS[1]);\n" +
                "                return spot - 1;\n" +
                "            end;\n" +
                "            return -1;";

        Long spot = (Long)client.eval(script, Collections.singletonList(key), Collections.emptyList());
        if (spot < 0) {
            System.out.println("failed to get the spot.");
            return false;
        } else {
            System.out.println("got one appointment spot.");
        }
        return true;
    } catch (Throwable throwable) {
        System.out.println("Failed to deduct spot: " + throwable.toString());
        return false;
    }
}
```

Message Queue (Kafka)

```java
@KafkaListener(topics = { "appointment" }, groupId = "group_1", containerFactory = "concurrentKafkaListenerContainerFactory")
public void onAppointmentCreationMessage (Appointment appointment) {
    // 1, parse request message of created appointment
    LOG.info("received request message of created appointment: : " + appointment.getAppointmentNo());
    appointment.setCreateTime(new Date());
    // 2, lock spot
    boolean lockSpotResult = appointmentInventoryDao.lockSpot(appointment.getAppointmentInventoryId());
    if (lockSpotResult) {
        // status of appointment: 0: no available spot, 1: appointment has been created, wait for confirmation
        appointment.setStatus(VALID);
        // user is added to limited user list
        redisService.addLimitedUser(appointment.getAppointmentInventoryId(), appointment.getUserId());
    } else {
        appointment.setStatus(INVALID);
    }
    // 3, insert appointment
    appointmentDao.insertAppointment(appointment);
}
```

Snowflake Algorithm

- Unique IDs that are sorted by time
- Can generate 69.73 years of IDs for twitter
- 41 bits – timestamp, 10 bits – machine, 12 bits per-machine sequence

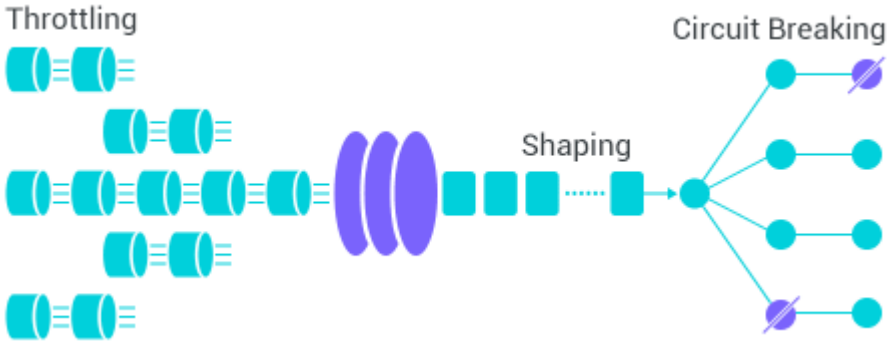
```
public Appointment createAppointment(int id, int userId) throws Exception {

    // 1. query & get appointmentInventory
    AppointmentInventory appointmentInventory = appointmentInventoryDao.queryAppointmentInventoryById(id);

    // 2. create & set new appointment information
    Appointment appointment = new Appointment();

    // use snowflake algorithm to generate appointment ID
    appointment.setAppointmentNo(String.valueOf(snowFlake.nextId()));
    appointment.setAppointmentInventoryId(appointmentInventory.getId());
    appointment.setUserId(userId);
    appointment.setAppointmentCount(APP_COUNT);
```

Sentinel Framework



- Protect key resources (appmt, schdle)
- Throttling and traffic shaping
- Defend system using circuit breaking
- Real-time monitoring and management

```java
public void appointmentsFlow(){
    //1, create a list to keep rules of rate limit
    List<FlowRule> rules = new ArrayList<>();
    //2, create main page rule to prevent visiting main page too fast
    FlowRule mainPageRule = new FlowRule();
    // - define protected resource
    mainPageRule.setResource("appointments");
    // - define rate limit type
    mainPageRule.setGrade(RuleConstant.FLOW_GRADE_QPS);
    // - define QPS
    mainPageRule.setCount(FAST_VISITS);

    // 3, create rule to limit number of visit of scheduling page
    FlowRule schedulingLimitRule = new FlowRule();
    schedulingLimitRule.setResource("scheduling");
    schedulingLimitRule.setGrade(RuleConstant.FLOW_GRADE_QPS);
    schedulingLimitRule.setCount(MAX_VISITS);

    // 4, all rules to the list
    rules.add(mainPageRule);
    rules.add(schedulingLimitRule);
    // 5, load the rules
    FlowRuleManager.loadRules(rules);
}
```

MySQL Database Tables

Appointment

Column	Type
appointment_count	int
appointment_inventory_id	int
appointment_no	varchar(100)
confirm_time	datetime
create_time	datetime
id	int
status	int
user_id	int

Appointment Inventory

Column	Type
active	int
available_spot	int
description	varchar(250)
end_time	datetime
id	int
locked_spot	int
name	varchar(250)
start_time	datetime
store_id	int
total_spot	int

Shop

Column	Type
address	varchar(250)
id	int
name	varchar(250)
phone	varchar(250)

Shopper

Column	Type
address	varchar(250)
id	int
name	varchar(250)
phone	varchar(250)

Thank You